STRCMP

Will fail if passed unterminated strings

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-17

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 3550 bytes

<table>
<tr><td><b>Attack Category</b></td><td>• Denial of Service</td></tr>
<tr><td><b>Vulnerability Category</b></td><td>• No Null Termination</td></tr>
<tr><td><b>Software Context</b></td><td>• String Management</td></tr>
<tr><td><b>Location</b></td><td></td></tr>
<tr><td><b>Description</b></td><td>strcmp and related functions will fail if passed unterminated strings.
strcmp() compares two strings. Like most C string functions, it relies on strings being null terminated.</td></tr>
<tr><td><b>APIs</b></td><td><table>
<tr><th>Function Name</th><th>Comments</th></tr>
<tr><td>lstrcmp</td><td></td></tr>
<tr><td>lstrcmpi</td><td></td></tr>
<tr><td>strcmp</td><td></td></tr>
<tr><td>strcmpi</td><td></td></tr>
</table></td></tr>
<tr><td><b>Method of Attack</b></td><td>strcmp() will fail if strings are not properly null terminated. This could allow an attacker to crash a program if he can force an unterminated string to be passed in.</td></tr>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
<tr><td><b>Solutions</b></td><td><table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>When one needs to compare strings.</td><td>Ensure that strings are null terminated before passing into strcmp. This can be enforced by always placing a \0 in the last allocated byte of the buffer.</td><td>Effective whenever one knows buffer size and so can add null.</td></tr>
</table></td></tr>
</table>

[1] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

<table>
<tr><td><b>Signature Details</b></td><td>int strcmp (const char * string1, const char * string2)</td></tr>
<tr><td><b>Examples of Incorrect Code</b></td><td>char str1[] ="something";
char str2[] = "another thing";
/* In this case we know strings are null terminated. Pretend we don't. Then the following is unsafe. */
if (strcmp(str1, str2)) { /* do something */ } else { /* do something else */ }</td></tr>
<tr><td><b>Examples of Corrected Code</b></td><td>char str1[] ="something";
char str2[] = "another thing";
/* In this case we know strings are null terminated. Pretend we don't. */
str1[sizeof(str1)-1] = '\0';
str2[sizeof(str2)-1] = '\0';
/* Now the following is safe. */
if (strcmp(str1, str2)) { /* do something */ } else { /* do something else */ }</td></tr>
<tr><td><b>Source Reference</b></td><td>• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vclib/html/_crt_strcmp.2c_.wcscmp.2c_._mbscmp.asp[2]</td></tr>
<tr><td><b>Recommended Resource</b></td><td></td></tr>
<tr><td><b>Discriminant Set</b></td><td><table><tr><td><b>Operating Systems</b></td><td>• Windows
• UNIX</td></tr><tr><td><b>Languages</b></td><td>• C
• C++</td></tr></table></td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com